INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

A Binary Keyword Spotting System with Error-Diffusion Based Feature Binarization

Dingyi Wang[1], *Mengjie Luo*[1,2], *Lin Li*[1,2], *Xiaoqin Wang*[1,2,*], *Shushan Qiao*[1,2,*], *Yumei Zhou*

[1]Institute of Microelectronics of the Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
{wangdingyi, luomengjie, lilin2020, wangxiaoqin, qiaoshushan, ymzhou}@ime.ac.cn

Abstract

Binary-neural-network based keyword spotting (KWS) for resource-constrained devices has gained much attention in recent years. Although several works proved their success, a fully binary KWS system is yet to come, considering high-precision speech feature maps are still required for satisfying accuracy. Such precision mismatch results in non-binary activation layers, thus leading to extra computational costs. In this paper, we present an extremely compact KWS system using a binary neural network and error-diffusion binarized speech features. The system eliminates all high-precision multiplications and requires only hardware-friendly bit-wise operations and additions for inference. Experiments on the Google speech commands show that our binary KWS system yields 98.54% accuracy on a 1-keyword task and 95.05% on a 2-keyword task, outperforming 8-bit KWS systems of bigger size. The result proves the feasibility of a fully binary KWS system and can be inspiring for hardware implementations.

Index Terms: Keyword spotting, binary neural network, error diffusion, convolutional neural networks

1. Introduction

Keyword spotting (KWS) is a critical component in low-power edge devices. Serving as an always-on audio switch to activate the whole system, it requires both low computation complexity and high accuracy. With remarkable progress in cognitive tasks, deep neural network (DNN) based approaches have become the major choice for building KWS systems[1]–[6]. In particular, convolutional neural networks (CNNs) are widely welcomed for high accuracy with less parameters[6]–[12]. Several novel network architectures such as DSCNN, TENet, TC-ResNet and MatchboxNet are proposed to further reduce computational cost for small-footprint KWS systems[13]–[16]. However, these networks are still too heavy for extreme low-power devices with limited computational and storage resources.

Recently, low-precision network quantization methods, especially the binary neural networks (BNNs) gain much attention in small-footprint neural network studies[17]–[20]. KWS implementations are also influenced by these techniques [21]. Several KWS application specific integrated circuit (ASIC) systems achieve stunning power efficiency with the help of BNN architecture[22][23]. Nevertheless, all these studies still require high precision speech features to maintain satisfying accuracy. Take [23] as an example, the processing element (PE) arrays designed for its proposed NN accelerator have to use a set of gates and registers to support 8-bit computing mode, while 8-bit computations are only executed in the first network layer. Such mismatch of precision between input features and network architecture results in extra resources and power, indicating there is still room for more compact KWS systems.

Although high precision speech features can lead to extra computational and storage cost, few works focus on low-precision speech representation approaches. Riviello and David[24] first proposed a log-Mel based linear quantizer that supports as low as 2 bits and shown insignificant accuracy loss on EdgeSpeechNet-A[25]. Cerutti et al.[26] proposed an MCU-based KWS system using analog binary features, while it is a simple threshold method and the system still uses high precision weights for the last layer of the neural network. Our previous work [27] proposed an error-diffusion based quantization, demonstrated that the bit-width of practical quantization can be further reduced to 1 bit. The method achieves state-of-the-art performance on KWS among all quantization methods and can accommodate different small-footprint network architectures, while the satisfying results are based on full-precision networks. All the fore-mentioned works haven't proved the feasibility of a fully binary KWS system which uses binary features and network with only binary activation and binary weights.

In this study, binary speech feature is combined with a fully binary neural network to produce a small-footprint KWS system. The contributions of our work are three-fold. First, the error-diffusion based binarization algorithm is optimized by introducing bit-shift operations to reduce computational complexity, especially from a hardware perspective. Second, a binary KWS network based on TC-ResNet8[14] is proposed, named as TC-BiReal8, with improved shortcut module employed for better performance. Third, the experiment on Google speech commands shows that by combining two techniques, our system achieves an accuracy of 95.05% when performing a 2-keyword task and 98.54% for a 1-keyword task, delivering better performance with fewer memory and 17x less energy cost than 8-bit systems. The result demonstrates that a fully binary NN system can deliver satisfying performance in small-footprint KWS tasks and can be helpful for future extreme low-power hardware implementations.

The rest of this paper is organized as follows: Section 2 briefly introduces the proposed feature binarization method and TC-BiReal8 KWS network. Section 3 presents the settings and results of the experiments. Section 4 concludes this paper.

2. Methods

2.1. Error-diffusion based feature binarization

The proposed binarization method is based on error-diffusion, which is a famous image processing algorithm for continuous

* Corresponding author

10.21437/Interspeech.2023-258

tone image quantization[28]. As the feature maps in CNN based KWS systems can be seen as images, image-based quantization methods can be used here and properly preserve the feature maps' information. The essence of the algorithm is shown in Fig.1, which is diffusing the quantization error from thresholding one pixel to its neighbors and thus influencing the neighbors' threshold operation. The algorithm starts from the top-left pixel of the image and runs firstly along the feature direction of feature maps and then along the time direction.

The diffusion kernel for error-diffusion method can be different in terms of size and values from application to application. Essentially, they can be treated as infinite impulse response (IIR) filters with slight difference on frequency response. For the sake of computational complexity, a 2-row and 3-column kernel as shown in Figure 1 is applied in this study, with the neighbor pixel i denoted as 0 to 3 influenced by pixel P. Figure 2 gives the commonly used coefficients for a 2-row and 3-column kernel.

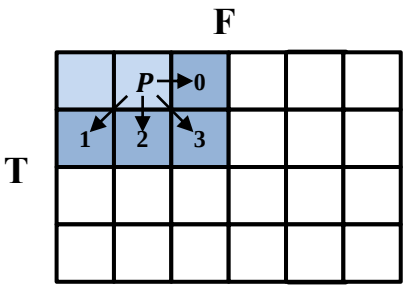

Figure 1: *Error-diffusion process. The quantization error of pixel P will be diffusing to its neighbor pixels numbered from 0 to 3.*

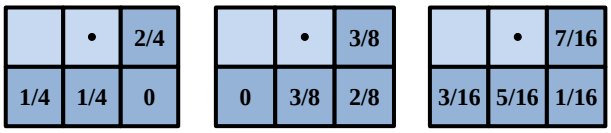

Figure 2: *Three 2-row-3-column Error-diffusion kernels. The quantization error will be multiplied by the coefficients at corresponding positions of the kernel before diffusion.*

Table 1: *Bit-shift numbers for different kernels.*

Position i \ Kernel k	a	b	c
0	$b^0 = 1$	$b^0 = 2$ $b^1 = 3$	$b^0 = 2$ $b^1 = 3$ $b^2 = 4$
1	$b^0 = 2$	-	$b^0 = 3$ $b^1 = 4$
2	$b^0 = 2$	$b^0 = 2$ $b^1 = 3$	$b^0 = 2$ $b^1 = 4$
3	-	$b^0 = 2$	$b^0 = 4$

Based on the fact that 8-bit feature is the common choice in state-of-the-art KWS ASIC implementations[22][23], this paper chooses the signed 8-bit fixed-point log-Mel (filter-banks) spectrogram as the quantization basis. Moreover, with signed fixed-point features, we can dramatically reduce the computational complexity of the error-diffusion binarization process from a hardware perspective. Noted that all the coefficients mentioned above are sums of different powers of 2, thus the computation results can be acquired by addition and bit-shift operations without multiplications. Assuming the quantization error is e, the diffusing error for neighbor position i is h_i, the nth bit-shift number of kernel k in position i is b_{ik}^n, and the array which stores N_k number of bits required for bit-shifting with kernel k is B_k, then we can derive that:

$$h_i = \sum_{n=1}^{N_k} (e \gg b_{ik}^n), b_{ik}^n \in B_k \quad (1)$$

where Array B_k is summarized in Table 1.

By using bit-shift operations, the proposed approach helps eliminate the multipliers in hardware and ultimately reduce computation cycles and extra-word-length registers required by multipliers. With adjustments mentioned above, the details of the error-diffusion binarization algorithm is summarized in Algorithm 1, operator $\gg$ refers to right-shifting.

Algorithm 1: Error-diffusion Based Feature Binarization

Input: $P_{in}(i,j) \in [-128,127]$ is the signed 8-bit input feature map value. F is the number of filter banks of feature maps while T is the number of time bins.

Output: $P_{out}(i,j)$ is the binary quantization value of feature maps.

for $i \leftarrow 1$ to T **do**
  for $j \leftarrow 1$ to F **do**
    $P_{out}(i,j) \leftarrow P_{in}(i,j) \gg 7$
    $e \leftarrow P_{in}(i,j)$ & $0x7F$
    if $(j < F)$ **then**
      $P_{in}(i,j+1) \leftarrow P_{in}(i,j+1) + \sum_{n=1}^{N_k} e \gg b_{0k}^n$
    end if
    if $(i < T)$ *and* $(j > 1)$ **then**
      $P_{in}(i+1,j-1) \leftarrow P_{in}(i+1,j-1) + \sum_{n=1}^{N_k} e \gg b_{1k}^n$
    end if
    if $(i < T)$ **then**
      $P_{in}(i+1,j) \leftarrow P_{in}(i+1,j) + \sum_{n=1}^{N_k} e \gg b_{2k}^n$
    end if
    if $(i < T)$ *and* $(j < F)$ **then**
      $P_{in}(i+1,j+1) \leftarrow P_{in}(i+1,j+1) + \sum_{n=1}^{N_k} e \gg b_{3k}^n$
    end if
  end for
end for

2.2. TC-BiReal8 binary neural network

We select TC-ResNet8 as our base network for binarization, considering it is a classic architecture that utilizes temporal convolution to achieve large receptive field for audio features and low computation complexity[15].

The basic topology of TC-ResNet8, including a basic convolution layer, 3 convolution blocks, an average pooling

layer and a fully connected layer, is retained in TC-BiReal8 except the activation and weights are binary now. However, we discover that a direct binarization of the network will suffer significant accuracy drop due to the low precision activation in shortcut convolutions. To mitigate the problem, a series of optimization enlightened by previous studies on BNN architectures[19][20] are introduced to the network. Firstly, this work transforms the original single shortcut connection to two shortcut connections in the temporal convolution block. Besides, an improved shortcut block, where feature maps of an 2x1 pooling layer and a 1-bit 1x1 convolution layer are concatenated on the channel dimension, is introduced to the convolution block to better preserve the representational capability of the down-sampling residual path. The adjusted convolution block is shown in Figure 3, and the topology of TC-BiReal8 is shown in Figure 4. By doing so, the proposed fully binary neural network can still deliver high performance under extreme compact size.

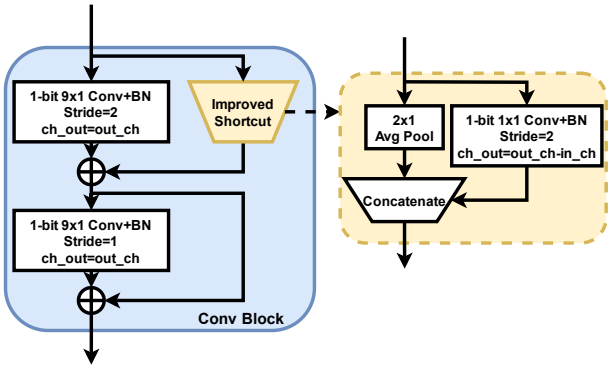

Figure 3: *Convolution block of TC-BiReal8. For the down-sampling phase, the shortcut is concatenated by the average-pooled input and feature maps output by 1x1 convolution block.*

Other than the convolution block optimization, we follow classic approaches of NN binarization techniques applied in the training stage to get a better KWS accuracy. Specifically, We use scaled weights proposed in [18] and $ApproxSign$ function proposed in [19] for activation with closer approximation of the non-differentiable $Sign$ function.

3. Experiments

3.1. Implementation details

3.1.1. Dataset and experiment setting

Google Speech Commands Dataset v1[29] is selected to conduct our following KWS experiments. The data set includes 30 target categories and contains 65k one-second-long utterances from 1881 speakers.

Our experiments are targeted on a 2-keyword task ("happy" and "stop") and a 1-keyword task ("happy") both with 2 extra classes ("silence" and "unknown") to simulate small-footprint KWS applications. The data set is split into training, validation, and test sets, with 80% training, 10% validation, and 10% test, respectively. To examine the performance, accuracy is used as the main metric.

Besides of the full-precision TC-ResNet8 and the binary TC-BiReal8, we also implement several baseline systems for comparison. A naïve binarized TC-ResNet8, denoted as TC-BiResNet8 for which we intuitively transform the convolution and fully-connected layers to binary ones without any topology changes, is used to validate the effectiveness of TC-BiReal8. Furthermore, two 8-bit DSCNNs which are about the similar memory consumption level of TC-BiReal8 are involved to evaluate our fully binary KWS system. The architecture of DSCNN is adopted from [13] but implemented with much shallower layers, including a basic 4x10 convolution layer, a 3x3 Depth-wise convolution layer, a 1x1 Point-wise convolution layer and a full connect layer, to match the memory difference.

3.1.2. Data augmentation and preprocessing

Following Google's preprocessing procedures, random shift and noise addition are applied as data augmentations to the training data. First, the background noises provided by the data set are sampled and added to the speech audio with a random proportion following a uniform distribution U (0, 0.1). Then the signal is time shifted by t seconds and zero-padded to 1 second, where t is sampled from U (−0.1, 0.1).

For raw feature extraction, 40 Mel filters are applied with a 30ms window size and a 10ms frame shift to generate feature maps with a size of 98x40, such setting can be seen in many small-footprint KWS applications[14]–[16]. For the simplicity of the experiments, the feature is min-max quantized to signed 8-bit fixed-point values before binarization.

3.1.3. Training Strategy

All the networks mentioned above are trained and evaluated in the experiments upon the PyTorch platform, with every model trained for 50 epochs with a batch size of 100, while the detailed training procedures are different. For TC-ResNet8 and DSCNNs, we follow the best-performing training hyper parameters and strategies listed in the original work except quantization-aware-training[30] is used for DSCNNs. For TC-BiReal8 and TC-BiResNet8, on the other hand, we use same training strategy, where Adam optimizer with a linear rate decay scheduler and an initial learning rate of 1e-2 is applied, and the weight decay is set to 5e-6 as suggested in [31]. All of the networks use cross-entropy loss for loss function.

3.2. Experimental results

3.2.1. Error-diffusion kernel

To start with, we conduct an experiment to investigate the difference among kernels mentioned in Section 2.1. This experiment targets on the two-keyword task and uses the full-precision model for fast evaluation. The result is summarized in Table 2.

Experiment shows that the best performing kernel (c) achieves an accuracy of 98.70%, only 0.18% lower than the 8-

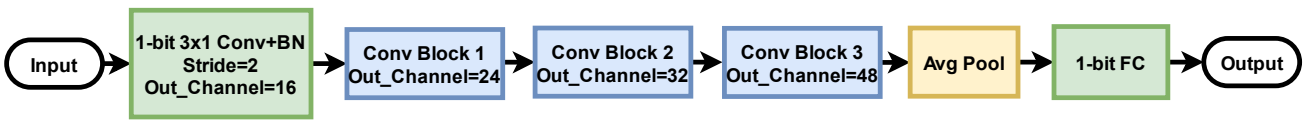

Figure 4: *Topology of TC-BiReal8.*

bit feature with 61K bit-wise OPs for error diffusion. Even kernel (a) achieves 98.19% accuracy with only 23K OPs. The result shows that binary representation only leads to an accuracy drop up to 0.69%. Besides, it also demonstrates that the accuracy difference on KWS among the kernels is no larger than 0.6%, indicating the choice of kernel has no significant influence on the performance of small-footprint KWS. Moreover, results show that the most complex kernel (c) will bring 2.03% extra OPs overhead. Even though the extra costs of all three kernels are quite small compared to the 3M OPs of network computation, the 0.77% OPs overhead of kernel (a) is clearly more attractive considering the little difference in KWS performance. Therefore, to make the best use of the computational cost advantage, we use kernel (a) for feature binarization in the rest of experiments.

Table 2: *Performance and bit-wise operations (OPs) for Error Diffusion of different kernels.*

Quantization method	Acc. (%)	Bit-wise OPs for Error Diffusion	Extra OPs Overhead*
8-bit log-mel	98.88	-	-
1-bit with Kernel (a)	98.19	23K	0.77%
1-bit with Kernel (b)	98.24	38K	1.26%
1-bit with Kernel (c)	98.70	61K	2.03%

*The overhead computation here ignores the complexity difference between bit-wise OPs and float OPs.

3.2.2. TC-BiReal8

To investigate the performance of TC-BiReal8, we conduct both the 1-keyword and 2-keyword experiment on the proposed TC-BiReal8 and naïve TC-BiResNet8 with compared to the full-precision TC-ResNet8. The results are shown in Table 3 and Table 4.

Table 3: *Performance of TC-BiReal8 on 1-keyword task.*

Network	Feature	Mem.	Speedup	Acc. (%)
TC-ResNet8	8-bit	264KB	1×	99.02
TC-BiResNet8	1-bit	8.2KB	~64×	96.25
TC-BiReal8	**1-bit**	**7.7KB**	**~64×**	**98.54**

Table 4: *Performance of TC-*BiReal8 *on 2-keyword task.*

Network	Feature	Mem.	Speedup	Acc.(%)
TC-ResNet8	8-bit	264KB	1×	98.88
TC-BiResNet8	1-bit	8.2KB	~64×	91.63
TC-BiReal8	**1-bit**	**7.7KB**	**~64×**	**95.05**

As the results show, TC-BiReal8 achieves 98.54% accuracy on the 1-keyword task and 95.05% on the 2-keyword task, which are both significant higher accuracy over the naïve binarized TC-BiResNet8 with a cost of slightly more additions but less parameters. Besides, the proposed fully binary system achieves insignificant accuracy drop as compared with the full-precision network on the 1-keyword task, while leads to 3.83% accuracy drop on the more complex 2-keyword task. Nevertheless, the binary system will save 32 times of storage and deliver at least 64 times of speedup over the full-precision one[32].

3.2.3. Comparison with 8-bit systems

To further validate the effectiveness of the fully binary system, TC-BiReal8 and two 8-bit DSCNN systems are examined on two KWS tasks. Table 5 and Table 6 give a brief summary of the results, where DSCNN-s refers to the DSCNN with 8 channels, and DSCNN-m refers to the one with 16 channels.

Table 5: *Comparison on 1-keyword task.*

Network	Feature	Total Mem.	Norm. Energy Saving	Acc. (%)
DSCNN-m	8-bit	16KB	1×	98.24
DSCNN-s	8-bit	10KB	2×	97.99
TC-BiReal8	**1-bit**	**8KB**	**17.1×**	**98.54**

Table 6: *Comparison on 2-keyword task.*

Network	Feature	Total Mem.	Norm. Energy Saving	Acc. (%)
DSCNN-m	8-bit	20KB	1×	93.07
DSCNN-s	8-bit	12KB	2×	90.56
TC-BiReal8	**1-bit**	**8KB**	**17.2×**	**95.05**

The experiment results demonstrate that our TC-BiReal8, using 1-bit feature, has a clear advantage over both the 8-bit systems on every perspective. The fully binary system surpasses DSCNN-s by 4.49% at most in 2-keyword task while consumes less total memory which includes the model size and feature storage. Besides, according to [33], the relative normalized energy cost ratio for operations of different bits can be estimated with approximated factors. Taking the energy cost for 8-bit multiplication and 8-bit addition are about 56 times and 8 times of energy cost for 1-bit operation, respectively, the binary system can achieve over 17 times of normalized energy saving compared to the 16-channel DSCNN-m.

4. Conclusions

In this paper, we aim to explore the potential of a fully binary KWS system with binary input features and a binary neural network for hardware friendly implementations. Error-diffusion feature binarization is employed and further optimized by replacing the multiplications with bit-shift operations. We also propose a binary neural network architecture named as TC-BiReal8, for which an improved shortcut technique is introduced to enhance the representation capacity of binary convolution blocks. Experiments demonstrate that by combining two techniques, our system achieves 98.54% accuracy for a 1-keywork task accuracy on Google speech commands v1, and 95.05% for a 2-keyword task. The binary system presents better performance than the 8-bit system of similar or slight bigger size. The results prove the feasibility of a fully binary KWS implementation and will be inspiring for extreme resource-limited devices and ultra-low-power ASIC deployment.

5. References

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4087–4091.

[2] M. Sun *et al.*, "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 474–480.

[3] S. Ö. Arık *et al.*, "Convolutional Recurrent Neural Networks for Small-Footprint Keyword Spotting," in *Proc. Interspeech*, 2017, pp. 1606–1610.

[4] R. Kumar, V. Yeruva, and S. Ganapathy, "On Convolutional LSTM Modeling for Joint Wake-Word Detection and Text Dependent Speaker Verification," in *Proc. Interspeech*, 2018, pp. 1121–1125.

[5] Y. Huang, T. Hughes, T. Z. Shabestary, and T. Applebaum, "Supervised Noise Reduction for Multichannel Keyword Spotting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5474–5478.

[6] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. Interspeech*, 2015, pp. 1478–1482.

[7] M. Yu, X. Ji, B. Wu, D. Su, and D. Yu, "End-to-End Multi-Look Keyword Spotting," in *Proc. Interspeech 2020*, 2020, pp. 66–70.

[8] O. Rybakov, N. Kononenko, N. Subrahmanya, M. Visontai, and S. Laurenzo, "Streaming Keyword Spotting on Mobile Devices," in *Proc. Interspeech*, 2020, pp. 2277–2281.

[9] R. Tang, W. Wang, Z. Tu, and J. Lin, "An Experimental Analysis of the Power Consumption of Convolutional Neural Networks for Keyword Spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5479–5483.

[10] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 416–421.

[11] A. Riviello, "Binary neural networks for keyword spotting tasks," Ecole *Polytechnique*, Montreal (Canada), 2020.

[12] I. López-Espejo, Z.-H. Tan, J. H. L. Hansen, and J. Jensen, "Deep Spoken Keyword Spotting: An Overview," *IEEE Access*, vol. 10, pp. 4169–4199, 2022,

[13] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *ArXiv Prepr. ArXiv171107128*, 2017.

[14] S. Choi *et al.*, "Temporal Convolution for Real-Time Keyword Spotting on Mobile Devices," in *Proc. Interspeech*, 2019, pp. 3372–3376.

[15] X. Li, X. Wei, and X. Qin, "Small-Footprint Keyword Spotting with Multi-Scale Temporal Convolution," in *Proc. Interspeech*, 2020, pp. 1987–1991.

[16] S. Majumdar and B. Ginsburg, "MatchboxNet: 1D Time-Channel Separable Convolutional Neural Network Architecture for Speech Commands Recognition," in *Proc. Interspeech*, 2020, pp. 3356–3360.

[17] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *ArXiv Prepr. ArXiv160202830*, 2016.

[18] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*, 2016, pp. 525–542.

[19] Z. Liu, W. Luo, B. Wu, X. Yang, W. Liu, and K.-T. Cheng, "Bi-real net: Binarizing deep network towards real-network performance," *Int. J. Comput. Vis.*, vol. 128, no. 1, pp. 202–219, 2020.

[20] J. Bethge, C. Bartz, H. Yang, Y. Chen, and C. Meinel, "Meliusnet: Can binary neural networks achieve mobilenet-level accuracy?," *ArXiv Prepr.* ArXiv200105936, 2020.

[21] R. Alvarez, R. Prabhavalkar, and A. Bakhtin, "On the Efficient Representation and Execution of Deep Acoustic Models," in *Proc. Interspeech*, 2016, pp. 2746–2750.

[22] B. Liu *et al.*, "A 22nm, 10.8 µ W/15.1 µ W Dual Computing Modes High Power-Performance-Area Efficiency Domained Background Noise Aware Keyword- Spotting Processor," *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 67, no. 12, pp. 4733–4746, 2020,

[23] W. Shan *et al.*, "A 510-nW Wake-Up Keyword-Spotting Chip Using Serial-FFT-Based MFCC and Binarized Depthwise Separable CNN in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, 2021,

[24] A. Riviello and J.-P. David, "Binary Speech Features for Keyword Spotting Tasks.," in *Proc. Interspeech*, 2019, pp. 3460–3464.

[25] Z. Q. Lin, A. G. Chung, and A. Wong, "Edgespeechnets: Highly efficient deep neural networks for speech recognition on the edge," *ArXiv Prepr. ArXiv181008559*, 2018.

[26] G. Cerutti, L. Cavigelli, R. Andri, M. Magno, E. Farella, and L. Benini, "Sub-mW Keyword Spotting on an MCU: Analog Binary Feature Extraction and Binary Neural Networks," *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 69, no. 5, pp. 2002–2012, 2022.

[27] M. Luo, D. Wang, X. Wang, S. Qiao, and Y. Zhou, "Error-Diffusion Based Speech Feature Quantization for Small-Footprint Keyword Spotting," *IEEE Signal Process. Lett.*, vol. 29, pp. 1357–1361, 2022.

[28] R. W. Floyd, "An adaptive algorithm for spatial gray-scale," in *Proc. Soc. Inf. Disp.*, 1976, vol. 17, pp. 75–77.

[29] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *ArXiv Prepr. ArXiv180403209*, 2018.

[30] B. Jacob *et al.*, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference." arXiv, 2017.

[31] Z. Liu, Z. Shen, S. Li, K. Helwegen, D. Huang, and K.-T. Cheng, "How *Do* Adam and Training Strategies Help BNNs Optimization?" arXiv, 2021.

[32] J. Bethge, H. Yang, M. Bornstein, and C. Meinel, "Binarydensenet: *developing* an architecture for binary neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, p. 0.

[33] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14.